# 1.0 Objectives

In this lab you will visualize the output of a 4-bit up counter on a 7-segment display with the help of a BCD-to-seven-segment decoder. This lab also covers the concept of debouncing and its importance in designing and debugging circuits with push buttons.

# 2.0 Parts List

# 

| **Quantity** | **Item** | |
| --- | --- | --- |
| 2 | White 830-point Breadboard | |
| Set of | Breadboard Wire Spools | Or, you could use a  Pre-Cut Wire Kit |
| 1 | Wire Cutters Electronic Grade |
| 1 | Wire Strippers Electronic Grade |
| 1 | 4-Position DIP Switch SPST (e.g., [5435640-2](https://www.digikey.com/en/products/detail/te-connectivity-alcoswitch-switches/5435640-2/969224) or [BPA04B](https://www.digikey.com/en/products/detail/c-k/BPA04B/949993)) | |
| 1 | 4-Bit Binary Ripple Counter ([CD74HCT93E](https://www.digikey.com/en/products/detail/texas-instruments/CD74HCT93E/38778)) | |
| 1 | BCD-to-Seven-Segment Decoder, Common-Anode ([SN7447AN](https://www.digikey.com/en/products/detail/texas-instruments/SN7447AN/1575197)) | |
| 1 | BCD-to-Seven-Segment Decoder, Common-Cathode ([SN74LS48N](https://www.jameco.com/Jameco/Products/ProdDS/47811.pdf)) | |
| 1 | Hex Schmitt-Trigger Inverters ([SN7414N](https://www.digikey.com/en/products/detail/texas-instruments/SN7414N/1575159)) | |
| 1 | Vertical 7-Segment Display, Common-Anode ([LSHD-5601](https://www.digikey.com/en/products/detail/liteon/LSHD-5601/560008?s=N4IgTCBcDaIIwDYAMBaOBWA7OlOByAIiALoC%2BQA)) | |
| 1 | Vertical 7-Segment Display, Common-Cathode ([LSHD-5503](https://www.digikey.com/en/products/detail/liteon/LSHD-5503/560009?s=N4IgTCBcDaIDIGUASARAtAVgwBgMwgF0BfIA)) | |
| 2 | Push Button ([TS02-66-70-BK-100-LCR-D](https://www.digikey.com/en/products/detail/same-sky-formerly-cui-devices-/TS02-66-70-BK-100-LCR-D/15634375?gclsrc=aw.ds&&utm_adgroup=&utm_source=google&utm_medium=cpc&utm_campaign=Pmax%20Shopping_Product_Passives%20Overstock&utm_term=&utm_content=&utm_id=go_cmp-21280451924_adg-_ad-__dev-c_ext-_prd-15634375_sig-Cj0KCQiA-5a9BhCBARIsACwMkJ6F4ps1_3kmZjGG0smqNkhF7Nm-4tiPpABBoxpw9auFvPX7PZAZlJ8aAnkaEALw_wcB&gad_source=1&gclid=Cj0KCQiA-5a9BhCBARIsACwMkJ6F4ps1_3kmZjGG0smqNkhF7Nm-4tiPpABBoxpw9auFvPX7PZAZlJ8aAnkaEALw_wcB&gclsrc=aw.ds)) | |
| 2 | 1 µF Electrolytic Capacitor ([50YXM1MEFR5X11](https://www.digikey.com/en/products/detail/rubycon/50YXM1MEFR5X11/11312838)) | |
| 7 | 330 Ω THT Resistor | |
| 4 | 1 kΩ THT Resistor | |
| 1 | Breadboard Power Supply (e.g., [YwRobot MB-V2](https://static.rapidonline.com/pdf/73-4538_v1.pdf)) | |

# 3.0 Background

As the name suggests, a 7-segment display has seven LED segments. Each of them is controlled with a separate input. We will use a decoder to convert a four-bit binary number into seven signals that will drive the 7-segment display. Specifically, we will use the SN7447A chip, which represents these numbers as shown in Figure 1. Other decoders, which are more expensive, represent the numbers from 10 to 15 as a single hexadecimal digit (A through F).

## 

Figure 1: Segment names and possible visualizations with the SN7447A chip. From the datasheet.

A 7-segment display can be classified as either common cathode or common anode. A common cathode display has the cathodes of all segments grounded. The display shows digits when a high signal is applied to a subset of the anodes. Likewise, a common anode display connects the anodes of all segments to the positive voltage. In this case, digits appear when a low signal is supplied to the corresponding cathodes.

Additionally, as shown in Figure 2, a 7-segment display can be either horizontal or vertical, depending on the placement of the pins. Each type has ten pins: seven for the inputs to the individual segments, one for the decimal point input, and two common anode/cathode pins. We will use vertical displays in today's lab.

| 1. Horizontal Display | 1. Vertical Display |
| --- | --- |

Figure 2: Pin locations for two types of the 7-segment displays. From the datasheets.

## 3.1 Counters and Flip-Flops

Flip-flops are essential memory elements in digital circuits that store a single bit of information. They differ from basic logic gates because they have a clock input that controls when their state changes. The most common types are SR, D, JK, and T flip-flops, each with unique characteristics and applications. Flip-flops are edge-triggered, meaning they change state on either the rising edge (low to high transition) or falling edge (high to low transition) of the clock signal.

When multiple flip-flops are connected in sequence, they can form a counter where each flip-flop represents a single bit. Counters can be asynchronous or synchronous. In an asynchronous counter the clock signal is applied only to the first flip-flop. Subsequent flip-flops toggle based on the output of the previous one in the sequence, creating a ripple effect. In a synchronous counter, the clock signal reaches all flip-flops simultaneously, reducing delay and making the counter more reliable for high-speed applications.

There are many different types of counters, such as up counters, down counters, and decade counters (which resets to 0 after reaching 9). Some counters use external clock control, while others incorporate internal logic to modify behavior. Counters can have additional logic, such as reset, enable, or the ability to preload specific values. By carefully selecting the type of flip-flop and clocking mechanism, engineers can design counters tailored for specific timing, counting, or sequencing applications.

This lab will use the CD74HC93 chip, which is shown in Figure 3. It is an asynchronous negative-edge triggered up counter, consisting of four flip-flops with two asynchronous resets (MR1 and MR2) and two clocks ( and ). Each flip-flop contributes one bit to the count value. MR1 and MR2 are ANDed together to reset the counter when both are brought high. This logic is shown in Figure 4 below. The counter can be adjusted to have three or four output bits. Since we will be using all four bits, the output must be fed back into . The external clock pulses must be provided on the input.

| Figure 3: Pin configuration of the CD74HC93 counter. From the datasheet. | Figure 4: Functional block diagram of the CD74HC93 counter. From the datasheet. |
| --- | --- |

# 4.0 Common-Anode 7-Segment Display

Your task is to connect a 4-position DIP switch to a decoder and visualize the output of the decoder on a vertical 7-segment display (common-anode).

## 4.1 Place the Main Components

Start by placing the DIP switch, decoder, and 7-segment display on the breadboard. Follow the convention of placing inputs on the left and outputs/displays on the right. Make sure the 7-segment display is oriented correctly with the decimal point in the lower-right corner. Connect the power and ground pins of the decoder using red and black wire, respectively. Connect the upper pins of the all four switches to the positive rail. Your setup should look similar to Figure 5 below.

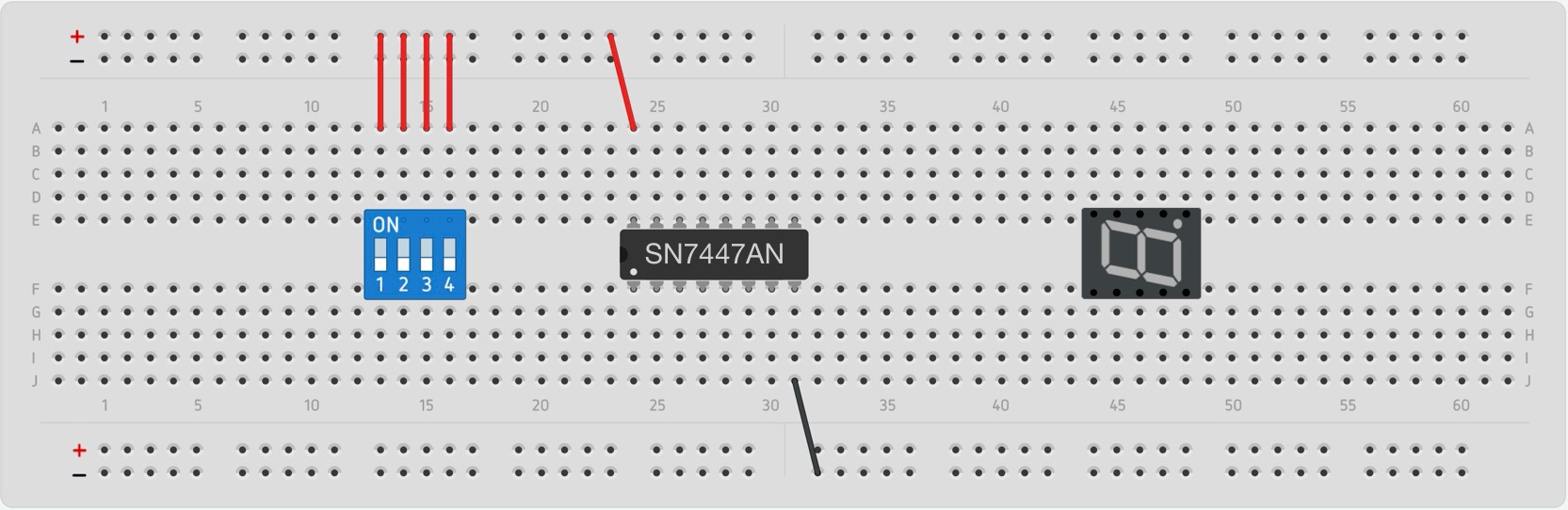


Figure 5: Place the 4-position DIP switch, decoder, and 7-segment display.

## **4.2 Wire the 4-Position DIP Switch**

Connect the lower pins of all switches to the negative rail using four 1 kΩ resistors. For this circuit, the left-most switch is the most significant bit (MSB) and the right-most switch is the least significant bit (LSB). In other words, the four inputs D, C, B, and A are mapped to switches 1, 2, 3, and 4 (in that order). The decoder uses the same ordering of its inputs (D, C, B, A), which map to pins 6, 2, 1, and 7, respectively. Make the appropriate connections as shown with blue wires in Figure 6. Also, connect pins 3, 4, and 5 of the decoder to the power rail using red wires.

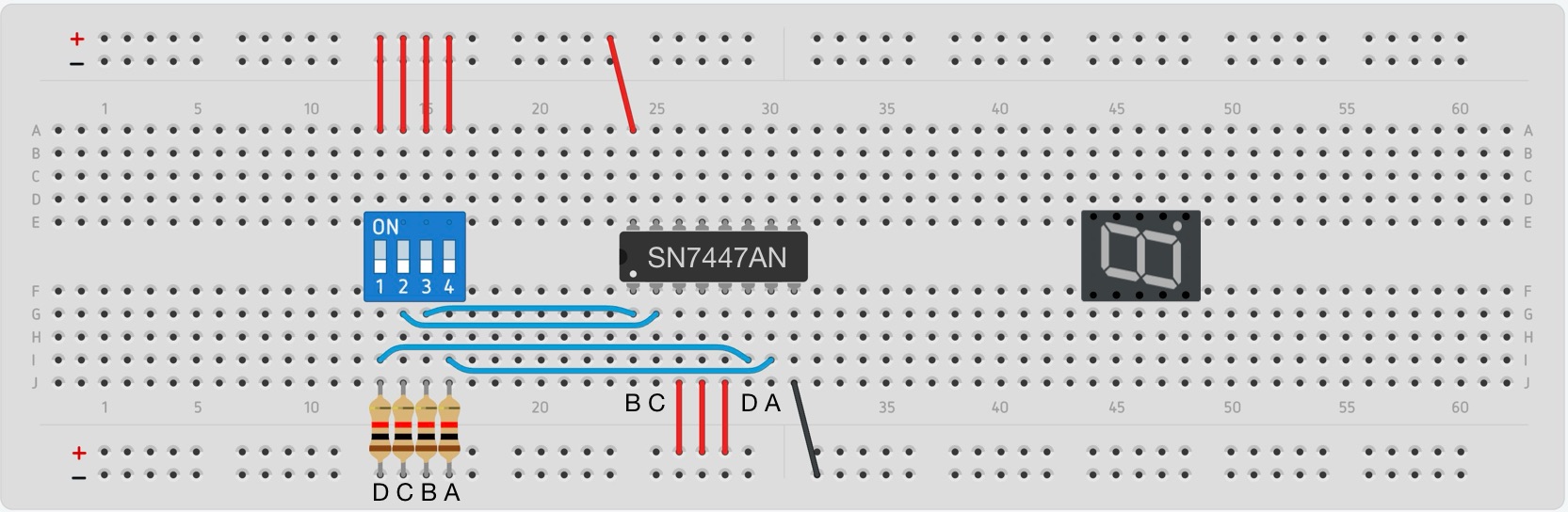


Figure 6: Connect the four switches to the inputs of the decoder.

## 

## **4.3 Wire the 7-Segment Display**

The main inputs of the 7-segment display are labeled A through G. The eighth input is the decimal point (DP), which is useful for some applications. The middle pins on both sides are reserved as common anodes; connect them high. The outputs of the decoder are labeled A through G (but with small letters) and correspond to pins 9 through 15 (though not in order). To finish the circuit, connect the outputs of the decoder, through 330 Ω resistors, to the corresponding pins of the 7-segment display (see Figure 7). Connect the decimal point high through a 330 Ω resistor as well, even though it it is not used here. The resistors are needed to balance the brightness of the LEDs when only a few of them are lit.

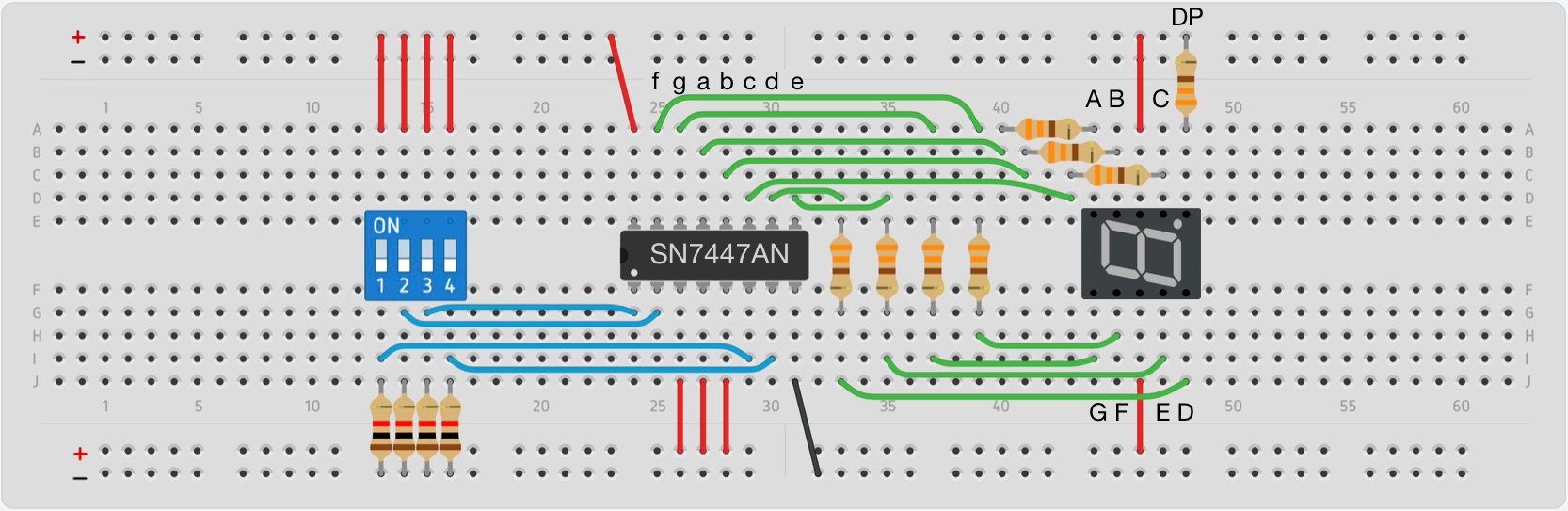


Figure 7: Connect the decoder to the display.

**4.4 Test the Display**

Your circuit should look similar to the real circuit in Figure 8. To test it, insert the power supply on the right side of the breadboard. Press the power button to turn it on, the display should light up. Using the four switches as input, test a few different values. Remember that values from 10 through 15 (in decimal) will not show up as A through F as they might on another display (Hint: check the spec sheet!). Show your results to the TA before proceeding to the next page.

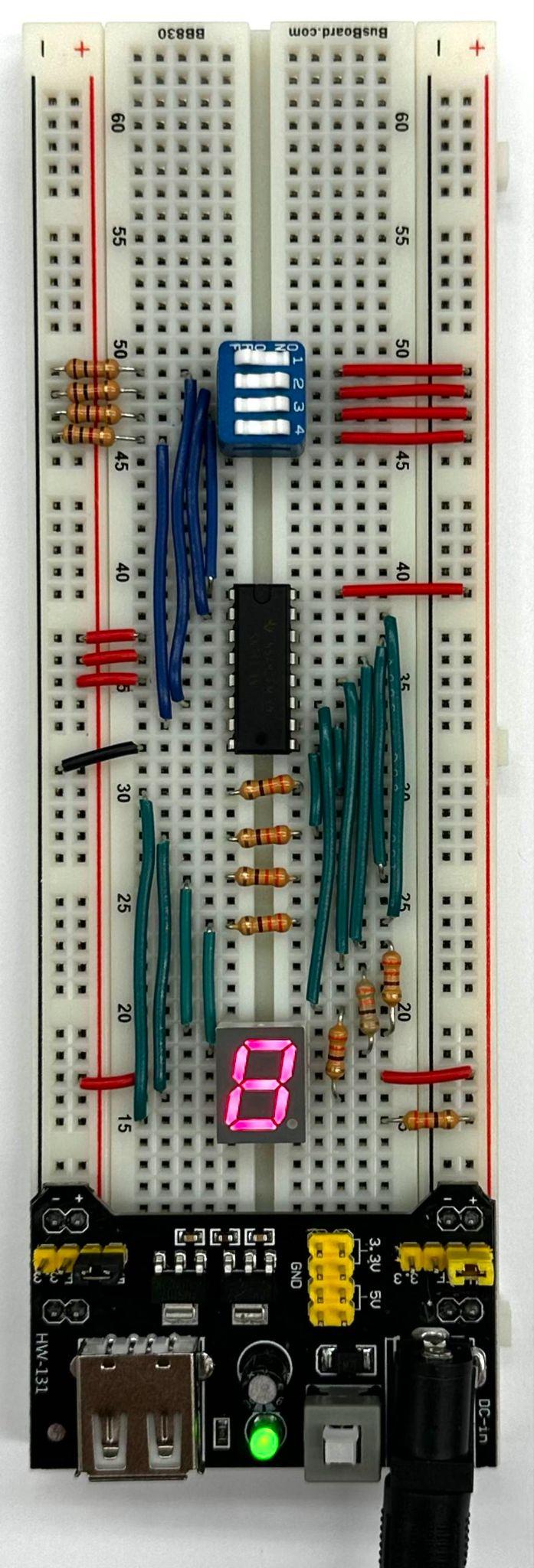
****

Figure 8: Example circuit connected to a breadboard power supply.

**5.0 Synchronous 4-Bit Up Counter**

Your next task is to replace the DIP switch with a 4-bit up counter. This counter will provide the four inputs to the decoder, which will then be visualized on the display. This will provide a certain level of automation, so you don’t have to input numbers manually.

## 5.1 Place the Up Counter

Remove the DIP switch (along with its corresponding wires and resistors) and replace it with the counter. Connect the chip to power and ground at pins 5 and 10, respectively (note: these pins are not in the typical places; see Figure 3). Connect output to input (pin 12 to pin 1) to configure the counter for 4-bit output. Lastly, connect MR1 to power and MR2 to ground (pins 2 and 3). To reset the counter, temporarily bring MR2 high (e.g., by moving the orange wire in Figure 9 to the positive rail).

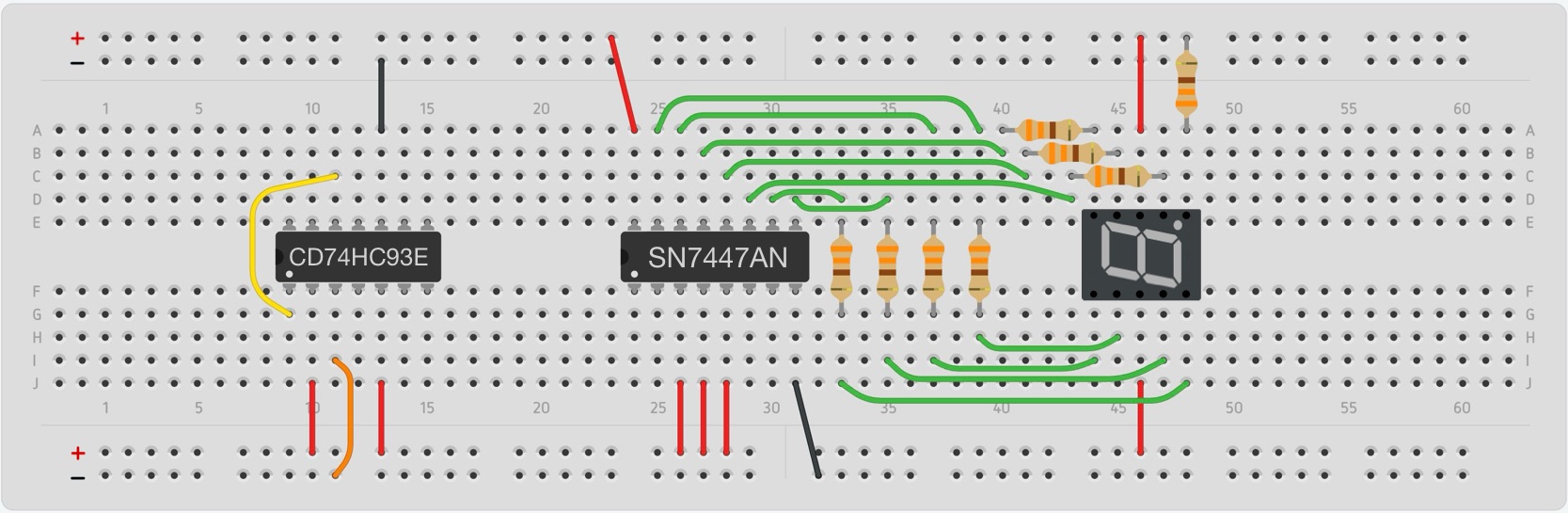


Figure 9: Replace the 4-position DIP switch with the 4-bit counter.

## 

## 5.2 Connect the Counter to the Decoder

Now that the counter is set up, connect the Q outputs of its flip-flops to the decoder. The least-significant bit is . The most-significant bit is . That is, connect output Q0 of the counter to input A of the decoder, output to input B, to C, and to D. Make these connections as indicated with the blue wires in Figure 10.

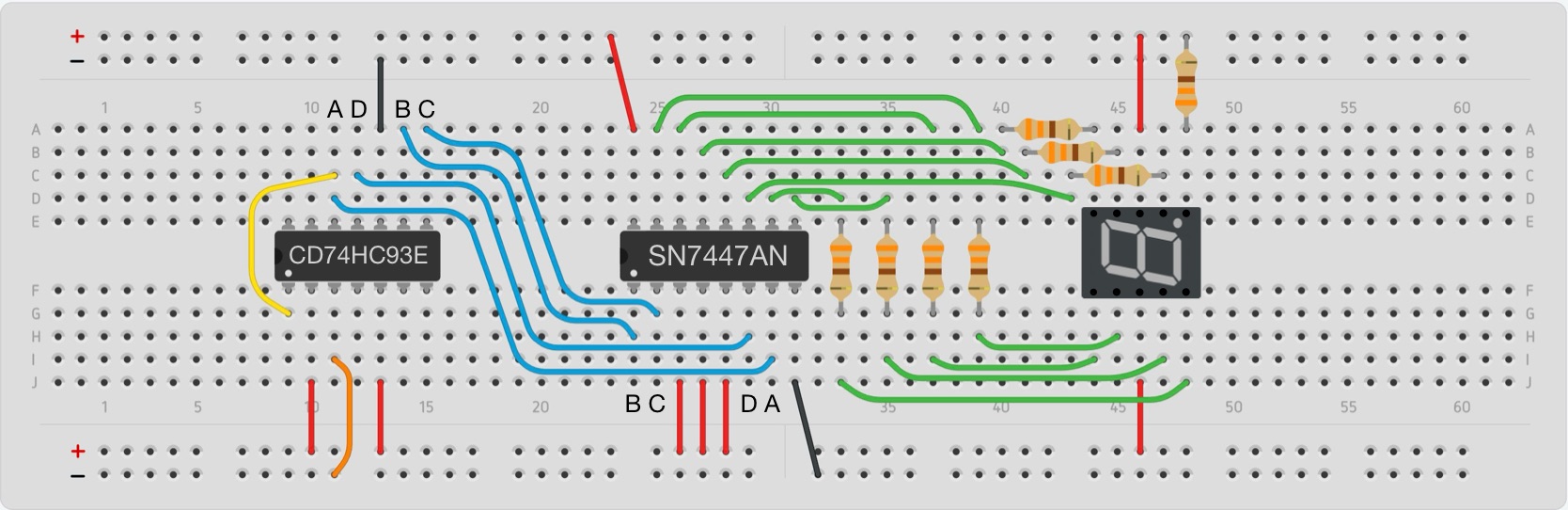


Figure 10: Connect the counter to the decoder.

## 5.3 Connect a Push Button as Clock

The counter requires an external clock, which will be provided with a push button. The push button has four pins that are connected to two separate terminals. When the button is pressed, the terminals are pushed together, closing the circuit and allowing current to flow. This provides a logical high or low signal depending on how the output wire is connected. When released, the button returns to its default open-circuit state.

Place a second breadboard below the first. Connect the power and ground rails of the two breadboards. Place a single push button towards the left side of the new breadboard, spanning the top and bottom portions. The top-right pin will be used as the input for the clock.

The counter chip uses negative-edge triggered flip-flops. Therefore, we want the counter to increment on a down press of the clock button. We need to wire the button such that the signal goes low when pressed and is high in the open-circuit state. To do this, you need to connect the terminal with the clock signal to Vcc and ground the opposite terminal. Connect the bottom-left pin of the button to ground and the upper-right pin to the positive rail through a 1 kΩ resistor. Then, connect the upper-right pin to pin 14 of the counter. The result should look similar to Figure 11.

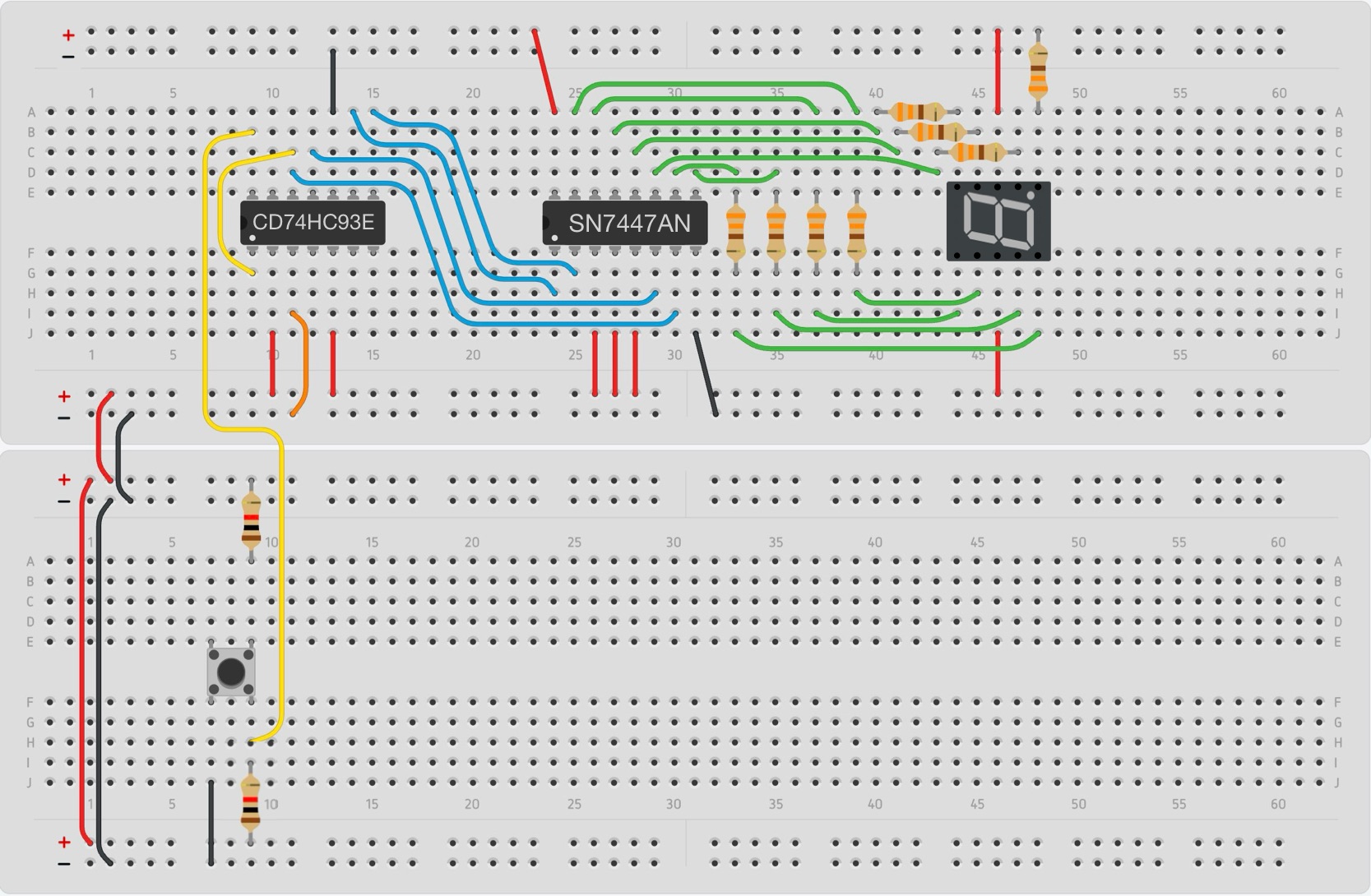
****

Figure 11: Connect a push button to the input of the counter.

**5.4 Test the Counter**

To test the counter, insert the power supply into the top breadboard and press the on button. To ensure that the count starts from zero, switch the orange wire that is connected to MR2 from ground to power and back. If connected properly, the display should count up from 0 to 15. Keep in mind that the numbers from 10 through 15 are displayed as shown in Figure 1.

As you test the circuit, you may notice that sometimes it increases the count by more than one for each press of the button. This is not the desired result and happens due to switch bouncing. We’ll fix this issue soon. For now, answer the questions for this section in the lab report and have the TA sign off on your results.

**6.0 Debouncing**

The undesired results from the previous section are due to switch bouncing. Bouncing is a phenomenon that occurs when a switch is pressed or released, causing unintended rapid transitions between on and off states before settling. This happens because the physical contacts within the switch do not cleanly make or break the connection. Switch bouncing can result in multiple unintended pulses being registered instead of a single, clean transition, which can cause erratic behavior.

Hardware-based solutions using capacitors and resistors or software-based approaches like time-delay filtering can ensure that only stable transitions are detected. This is known as debouncing. For this lab, we will use a Schmitt trigger inverter and an electrolytic capacitor to debounce the signal from a push button. When the button is pressed or released, the Schmitt trigger ensures that small fluctuations caused by bouncing do not immediately toggle the output. Instead, it only switches states when the input crosses well-defined high and low voltage thresholds. Additionally, the electrolytic capacitor creates a low-pass filter when the button is not pressed, which further helps to block any unintentional high signals. This effectively filters out rapid unwanted transitions, ensuring the up counter receives only a single clock pulse per button press.

## 6.1 Debounce a New Push Button

Place the Schmitt trigger inverter in the middle of the bottom breadboard. Connect it to power and ground. Add another push button to the right of the Schmitt trigger. Configure it similarly to the previous one but flipped since the signal is going to be inverted by the Schmitt trigger.

To accomplish this, connect the upper-right pin to Vcc and the bottom-left pin to ground with a 1 kΩ resistor. In addition, place an electrolytic capacitor between Vcc and the upper-left pin of the push button. Because this type of capacitor has a polarity, ensure that the positive pin (the longer one) is connected to power and the negative pin is grounded. Connect the upper-left pin of the button to the Schmitt trigger at pin 9. Then, connect pin 8 of the Schmitt trigger as the clock signal of the counter (i.e., pin 14 of the counter chip). The final circuit should look similar to Figure 12.

Now, when you test this circuit, you should see the display counting up consistently and incrementing only once each time the button is pressed and released. Before continuing, answer the questions for this section in the lab report.

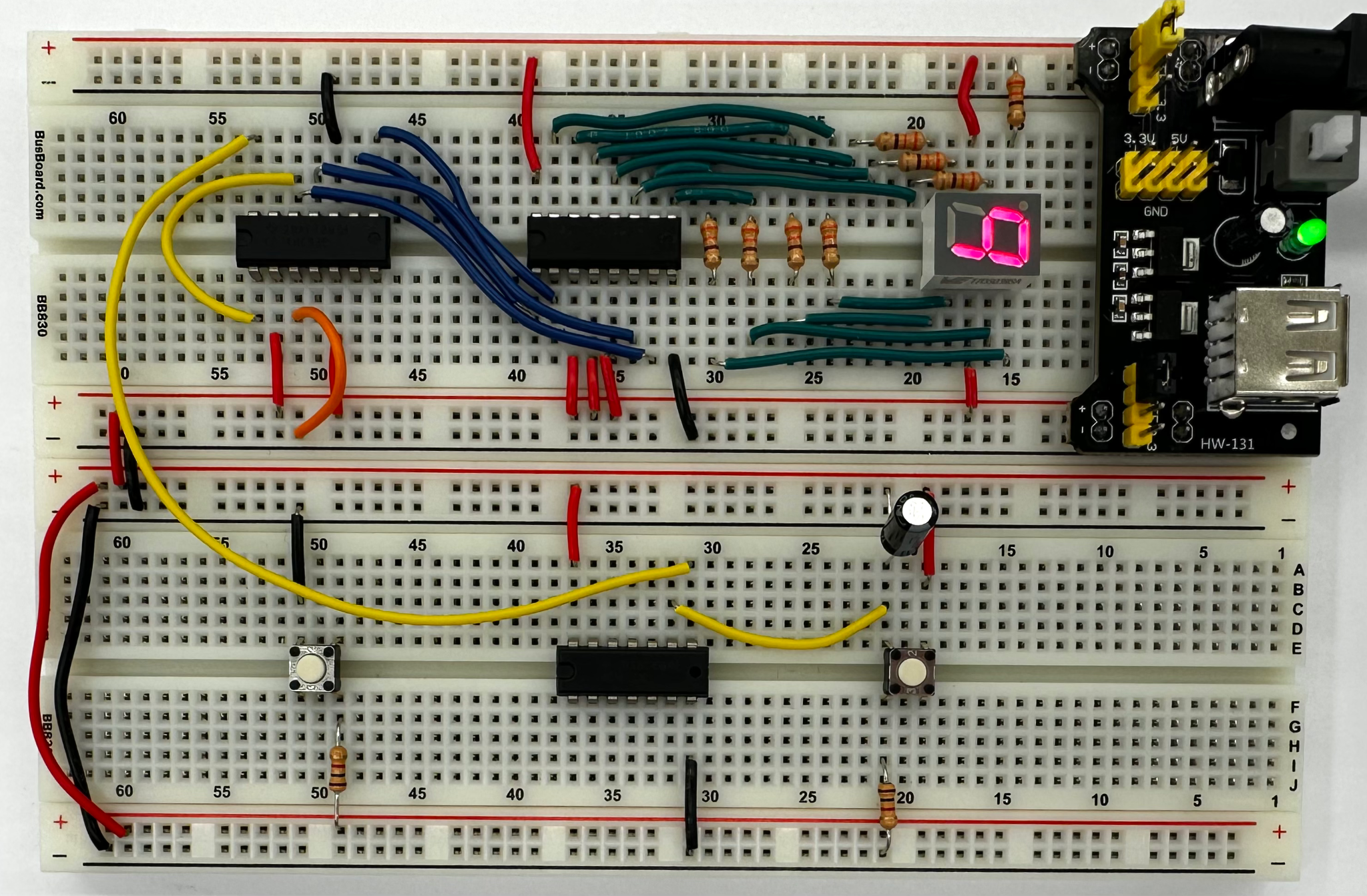


Figure 12: Circuit with debounced clock signal.

## 

## 6.2 Debounce the Original Button

To check your understanding of the Schmitt trigger specifications, your next task is to debounce the original push button. There are multiple inverters embedded in the Schmitt trigger chip and any of them can be used for this step.

Start by opening the datasheet, which can be found [here](https://www.ti.com/lit/ds/symlink/sn7414.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1740649737104&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253FdistId%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Fsn7414). Find the figure with the pin layout and use it to locate another inverter on the chip. Place a wire from the upper-right pin of the push button to the input of the inverter. Place a second wire from the output of the inverter to the input of the counter. Note: because the Schmitt trigger inverts the signal, the counter should now increment when the push button is released.

Once done, answer the questions in the lab report. Then, demonstrate **both** debounced buttons to the TA.

**7.0 Decade Counter**

The concept of a decade counter was briefly mentioned in the background section. A decade counter is a four-bit up counter, but instead of counting all the way to 15, it only counts up to 9 before resetting to 0. Thus, it only counts through ten digits, which explains its name.

Earlier, you connected MR1 to Vcc and MR2 to ground so the counter could be manually reset when the MR2 signal is brought high. Now, the challenge is to automatically reset the digit counter after the nine is reached. More specifically, it *should* display nine for a full clock cycle, but when it is incremented again, it should start back at zero. Keep in mind that this chip uses an asynchronous reset for an asynchronous counter.

To get started, think about what should cause this reset to happen. Also, consider which outputs could be used to trigger that transition and what connections you can add to make this happen. Once you have successfully completed this task, demonstrate your working solution to a TA.

**Extra Credit: Switch to a Common-Cathode 7-Segment Display**

Your last task is to modify the circuit to use a different 7-segment display. This new display is also vertical but uses a common cathode. In order to utilize this display, you will also need a new decoder that is compatible with common-cathode displays. In other words, you must replace the decoder and the display with the new components. Then, wire them based on their datasheets, which can be found in the parts list at the beginning of this document.